
Igra 1

TEME

- Pravljenje igrica i vezbanje prethodnog gradiva

PRIMER 1: Napisati program kojim se pravi +-e broja. Korisnik moze da bira svojim unosom tok igre, prekid, novu partiju, kraj programa. Nova partija podrazumeva da racunar bira proizvoljan broj od 1 do 100. Korisnik treba da pogodi, pri cemu se broje iteracije.

KORACI RESAVANJA:

- V1
 - Izbor slucajne vrednosti
 - Izbor korisnika
 - Detekcija pogotka ili promasaja
 - Testiranje
- V2
 - Pravljenje prethodnog da se ponavlja dok se ne pogodi
 - Testiranje
 - Dodavanje brojanja iteracija
 - Testiranje
- V3 - Finalna implementacija
 - Pravljenje da se kontrolise unos korisnika i da se na taj nacin bira nova igra, prekid trenutne igre ili kraj programa
 - Igranje igre ide u odvojenu funkciju

V1 IMPLEMENTACIJA:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    /* Deklaracija potrebne promenljive. */
    int slucajanBroj = (rand() % 100) + 1;
    int brojUnos;

    /* Ucitavanje vrednosti broja x. */
    printf("Unesite broj: ");
    scanf("%d", &brojUnos);

    /* Ispis rezultata. */
    if (brojUnos == slucajanBroj) {
```

```

    printf("Bravo, pogodio si!\n");
} else if (brojUnos < slucajanBroj) {
    printf("Trazeni broj je veci!\n");
} else {
    printf("Trazeni broj je manji!\n");
}
return 0;
}

```

V2 IMPLEMENTACIJA:

```

#include <stdio.h>
#include <stdlib.h>

int main() {
    /* Deklaracija potrebne promenljive. */
    int slucajanBroj = (rand() % 100) + 1;
    int brojUnos = -1;
    int brojIteracija = 0;

    while (brojUnos != slucajanBroj) {

        /* Ucitavanje vrednosti broja x. */
        printf("Unesite broj: ");
        scanf("%d", &brojUnos);
        brojIteracija++;

        /* Ispis rezultata. */
        if (brojUnos == slucajanBroj) {
            printf("Bravo, pogodio si iz %d pokusaja!\n", brojIteracija);
            break;
        } else if (brojUnos < slucajanBroj) {
            printf("Trazeni broj je veci!\n");
        } else {
            printf("Trazeni broj je manji!\n");
        }
    }
    return 0;
}

```

FINALNA IMPLEMENTACIJA:

```

#include <stdio.h>
#include <stdlib.h>

void igrajIgru() {
    /* Deklaracija potrebne promenljive. */
    int slucajanBroj = (rand() % 100) + 1;
    int brojUnos = -1;

```

```

int brojIteracija = 0;

while (brojUnos != slucajanBroj) {

    /* Ucitavanje vrednosti broja x. */
    printf("Unesite broj: (0 za prekid igre)");
    scanf("%d", &brojUnos);
    brojIteracija++;

    /* Ispis rezultata. */
    if (brojUnos == 0) {
        printf("Igra je prekinuta!\n");
        return;
    } else if (brojUnos == slucajanBroj) {
        printf("Bravo, pogodio si iz %d pokusaja!\n", brojIteracija);
        break;
    } else if (brojUnos < slucajanBroj) {
        printf("Trazeni broj je veci!\n");
    } else {
        printf("Trazeni broj je manji!\n");
    }
}
return;
}

int main() {
    printf("Igramo igru pogadjanja broja!\n");
    char unos;

    while (1) {
        printf("Za novu igru unesite n, za prekid programa unesite q.");
        scanf("%c", &unos);
        if (unos == 'n') {
            igranjIgru();
            continue;
        } else if (unos == 'q') {
            printf("Prekid programa!\n");
            break;
        } else if (unos != 'n' | unos != 'q'){
            printf("Neispravan unos!\n");
        }
    }
}

```

OBJASNJENJE

- Objasniti random
- Objasniti da funkcija moze da ima 2 ili vise return naredbi

- $num = (rand() \% (upper - lower + 1)) + lower$ Ako su upper i lower brojevi za opseg
- Scanf nekad ne uzme input. Smisliti jos neki nacin kako da se to ne desava

PRIMER 2: Pogadjanje zbira bacenih kockica, ako jednu baca korisnik a drugu baca racunar. Racunar baca kockicu i dobija random broj od 1 do 6. Korisnik unosi vrednost za svoju kockicu. Nakon toga, racunar pogadja zbir te dve kockice, igrac takodje pogadja zbir te dve kockice, ko pogodi dobija poen, a ako su oba igraca pogodila tacan zbir, niko ne dobija poen. Igra se igra do 3 dobijenih poena.

KORACI RESAVANJA:

- Ovo ucenik moze da radi
- V1
 - Jedan potez igre sa svim neophodnim promenljivima i racunanjem randoma
- V2
 - Da se potezi ponavljaju dok se ne dodje do 3 poena

FINALNA IMPLEMENTACIJA:

```
#include <stdio.h>
#include <stdlib.h>

int kockica() {
    return (rand() % 6) + 1;
}

void igranjIgru() {
    int racunarBroj = 0;
    int poeniRacunar = 0;
    int poeniIgrac = 0;
    int igracBroj = 0;
    int zbirRacunar = 0;
    int zbirIgrac = 0;
    int tacanZbir = 0;

    while (poeniRacunar < 3 & poeniIgrac < 3) {

        printf("Igrac  %d::%d  Racunar\n", poeniIgrac, poeniRacunar);
        racunarBroj = kockica();

        /* Ucitavanje vrednosti broja x. */
        printf("Unesite broj kockice: (0 za prekid igre)");
        scanf("%d", &igracBroj );
        if (igracBroj == 0) {
            printf("Igra je prekinuta!\n");
            return 0;
        }
    }
}
```

```

tacanZbir = igracBroj + racunarBroj;
printf("Unesite broj zbira: (0 za prekid igre)");
    scanf("%d", &zbirIgrac );
if (igracBroj == 0) {
    printf("Igra je prekinuta!\n");
    return 0;
}
zbirRacunar = kockica() + racunarBroj;
/* printf("%d %d\n", racunarBroj, zbirRacunar); */
/* Ispis rezultata. */
if (zbirRacunar == tacanZbir & zbirIgrac == tacanZbir) {
    printf("Nema poena!\n");
} else if (zbirRacunar == tacanZbir) {
    printf("Racunar dobija poen!\n");
    poeniRacunar++;
} else if (zbirIgrac == tacanZbir) {
    printf("Igrac dobija poen!\n");
    poeniIgrac++;
} else {
    printf("Nema poena u ovoj rundi!\n");
}
}
if (poeniRacunar == 3) {
    printf("Racunar je pobedio!\n");
}
if (poeniIgrac == 3) {
    printf("Igrac je pobedio!\n");
}
}

int main() {
    printf("Igramo igru pogadjanja zbira kockica!\n");

    while (1) {
        printf("Za novu igru unesite n, za prekid programa unesite
q.");
        char unos;
        scanf("%c", &unos);
        if (unos == 'n') {
            igrajIgru();
        } else if (unos == 'q') {
            printf("Prekid programa!\n");
            break;
        } else {
            printf("Neispravan unos!\n");
            continue;
        }
    }
}

```